

Sustainment

- Best practices/tools:

Inside HPC

- **Separate prototype development from production development**
- Sustainable software that can absorb good research ideas. Make this a process. (Visit/ MPICH/ACTS collection)
- Testing at scale - schedule the time because vendors/open-source don't have the resource
- Risk analysis – identify consequences of failure (vendor based, other)
- **Include SW capability and support in the system RFP (PAPI, IEEE floating point ...) to garner vendor support (1)**
- Multi-organization contracts – can we expand, what is the future model as we include other organizations (HPSS) (2)
 - Inventory software, fund according to weighted importance to labs
 - Sustainment reviews should be done by independent eyes
 - Funding models

Outside HPC

- Open-source groups that have a good quality gatekeeper
- There are model open source communities
- External vendors have separate organizations that support tools

Edu/Training

- Letting other know capabilities, expanding user community
- Gather from user community – what should change to engage and what should continue to be sustained – feedback loop
- Transitioning tools that will

- Worst practices:

- Wacky licensing. use standard open source licenses. Also provides a backup plan.
- Unhappy success. Management doesn't want to pay for support of external users
- Duplicate support contacts at a small scale, often duplicating scope To avoid, look at HPSS funding model
- Prototypes that are not ready to move toward production

- Top challenges?
 - Cross-cutting: providing hardware for testing at scale
 - Market may drive vendors and open-source communities away from HPC requirements
 - Prioritizing SW sustainment is hard
 - Research funding doesn't pay for sustainment
 - Since SW support funding is categorized under facilities – it competes with HW purchases
- New technologies required?
 - Facilities funding for software
 - Cross-org coordination process (inventory of strategic tool needs, coordination of support)
 - Single external org to coordinate funding
 - Is a facility approach needed to build support structure for tools. (virtual sw community)
 - Software dependencies – how do we better manage tests that can provide this information
- Worst practices:
 - Wacky licensing. use standard open source licenses. Also provides a backup plan.
 - Unhappy success. Management doesn't want to pay for support of external users
 - Duplicate support contacts at a small scale, often duplicating scope To avoid, look at HPSS funding model
 - Prototypes that are not ready to move toward production